
RoboTools

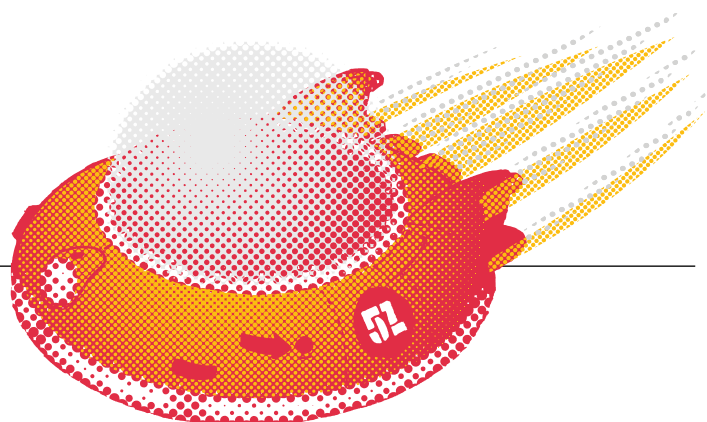
L'influenza dell'open source sugli strumenti di produzione del carattere tipografico

The influence of open source on typeface production tools

— English text on p. 37

Testo di Roberto Arista

Nel corso degli ultimi venti anni un gruppo di progettisti ha sistematicamente messo in discussione l'offerta sul mercato di **strumenti per il type design**. Un invito a riappropriarsi dei mezzi di produzione di lavoro unendo licenze retail e open source.



UF02
ROBOFAB.COM ROBOTHON09

Nel marzo 2015 il quieto caos che anima l'Accademia di Belle Arti dell'Aia subisce una lieve increspatura per un paio di giorni, è ora del Robothon, la conferenza dedicata allo sviluppo di strumenti per la progettazione tipografica che l'accademia ospita ogni tre anni.

L'atmosfera è quella delle riunioni di famiglia: c'è un po' tutta l'Olanda e la Berlino del type design più una discreta delegazione dalle coste Ovest ed Est degli Stati Uniti. Dopo il rituale giro di convenevoli, la folla sciamma verso l'auditorium, pronta ad immergersi in un fitto

programma di presentazioni. La conferenza viene aperta dai saluti di Guido van Rossum, *benevolo dittatore a vita* di Python e tra i nomi più influenti dell'informatica contemporanea. È una storia che inizia una ventina di anni prima composta da fratelli, compagni d'accademia e ibridazione fra *open* e *retail*.

Nei primi anni Novanta la rivoluzione del desktop publishing è in atto ormai da una decina di anni e le font digitali sono ancora uno dei pilastri fondanti della democratizzazione informatica. Dai costosi e ingombranti sistemi di digitalizzazione si è passati stabilmente ad applicativi desktop in grado di supportare il disegno di curve di Bézier cubiche e l'esportazione in formato PostScript. L'applicativo più diffuso tra i professionisti del settore è Fontographer, software in produzione dalla metà degli anni Ottanta dal cui nocciolo prenderà vita il noto FreeHand.¹ Nel tentativo di combattere il monopolio sul formato PostScript di Adobe, Apple e Microsoft rilasciano nel 1991 il formato TrueType, tra le cui peculiarità vi è il fatto di sfruttare curve di Bézier quadratiche, che non è possibile convertire senza perdita di qualità o aggiunta di punti da quelle di terzo grado. Per le compagnie che vogliono operare con un controllo totale sul disegno, diventa quindi necessario manipolare direttamente nell'editor le nuove curve. Su iniziativa di David Berlow, fondatore di FontBureau, Petr van Blokland insieme ai Letterror – suo fratello minore Erik e Just van Rossum – distribuiscono nel 1996 una versione derivata di Fontographer, RoboFog.

Più si procede con il disegno di un carattere, più le operazioni ripetitive crescono esponenzialmente: che sia il posizionamento di segni diacritici o l'interpolazione di più master, sviluppare un'interfaccia per eseguire operazioni così diverse appare rapidamente un esercizio non praticabile. Nasce quindi l'idea di creare una porta di accesso alle funzionalità di base dell'applicativo mediante l'implementazione di un linguaggio di scripting.

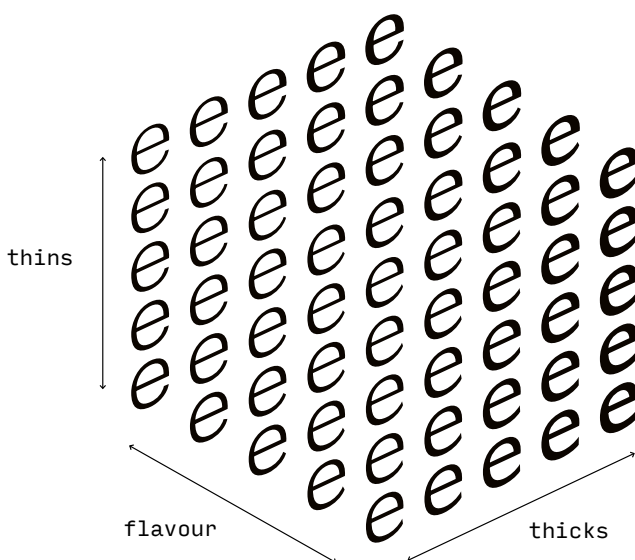
Dopo un primo tentativo con AWK, Just propone di dare una possibilità al linguaggio open source sviluppato dal fratello maggiore Guido, la cui prima versione era del gennaio 1994. È un'unione felice, destinata ad influenzare radicalmente il font editing: Python, progettato per un pubblico di non programmatori, presenta infatti una sintassi molto asciutta e accessibile a scapito della velocità di calcolo. RoboFog viene sviluppato fino al 2001, momento in cui il vecchio codice sorgente di Fontographer non sopravvive al passaggio a OSX.

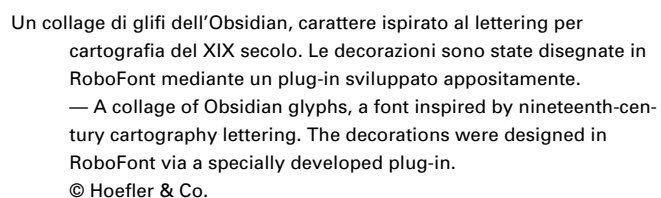
Già nel 1998 Yuri Yarmola, sviluppatore di FontLab, aveva integrato Python all'interno della versione 2.0 dell'applicativo rilasciando un'API (Application Program Interface) Python chiavi in mano. Quando è terminato lo sviluppo di RoboFog e i suoi utenti sono migrati verso FontLab, i Letterror, con l'aiuto di Tal Leming, hanno usato questa API per trasferire il modello a oggetti di RoboFog nel nuovo applicativo. Lo sviluppo di questo modulo è iniziato nel 2003 e il risultato è stato presentato un anno più tardi con il nome di RoboFab. Il modulo introduce un nuovo formato font chiamato UFO (Unified Font Object), oggi formato standard per l'intercambio di dati font. La filosofia progettuale su cui si basa è riassunta in forma di manifesto:

1. I dati devono essere leggibili ed editabili da parte degli utenti.
2. I dati devono essere indipendenti dall'applicazione.
3. La duplicazione dei dati deve essere evitata a meno che sia assolutamente necessaria.

L'identità di Robothon09. — Identity of Robothon09.
© Robofab Consortium

Il celebre cubo di Noordzij è il modello che ha ispirato MutatorMath e SuperPolator di Erik van Blokland. L'immagine è stata realizzata dall'autore a partire da tracciati disegnati da Gerrit Noordzij mediante MutatorMath e Drawbot.
© Originariamente disegnato da Gerrit Noordzij intorno al 1973, tracciati digitalizzati per Ikarus e rifiniti da Petr van Blokland intorno al 1985. Ridigitalizzati dall'opera *Ikarus* di Petr van Blokland.
— The famous Noordzij cube is the model that inspired MutatorMath and SuperPolator by Erik van Blokland. The image was created by the author from outlines designed by Gerrit Noordzij using MutatorMath and Drawbot.
© Originally drawn by Gerrit Noordzij around 1973, digitized for Ikarus and refined by Petr van Blokland around 1985. Redigitized from Petr's Ikarus artwork.





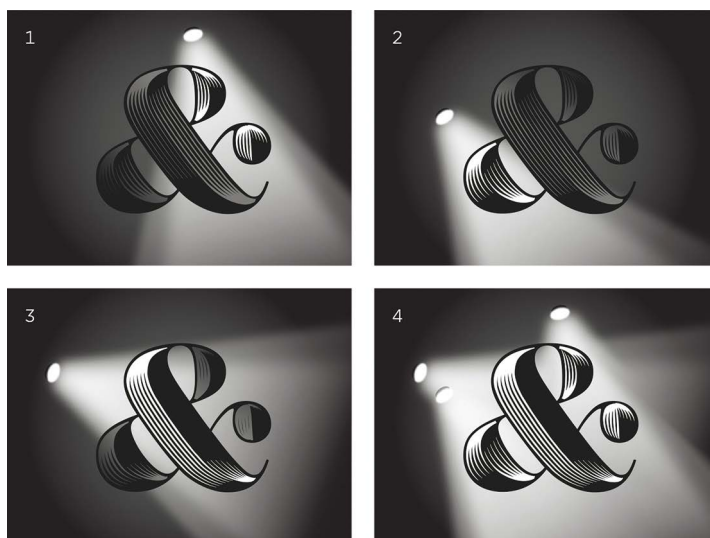
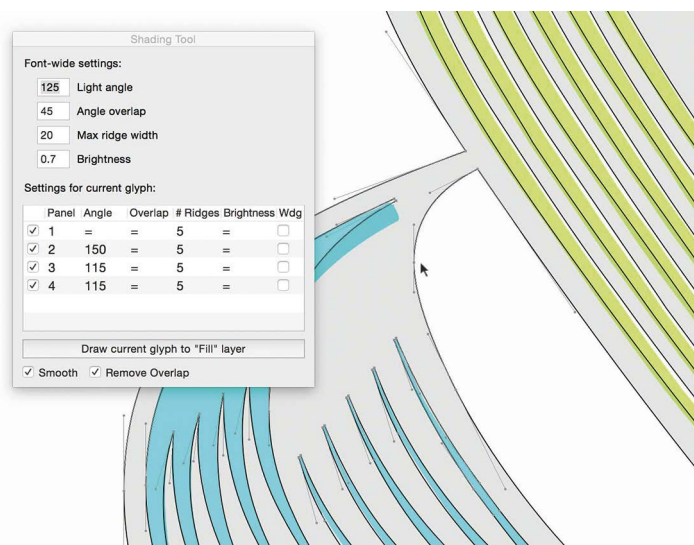
Nei primi anni Duemila il mercato è dominato da FontLab, che però non riesce a tenere il passo con i nuovi bisogni progettuali. FontLab, infatti, è progettata per essere un'applicazione generalista, non necessariamente orientata verso un pubblico specializzato. Nel frattempo alcuni dei progettisti incontrati sinora proseguono nello sviluppo di applicativi in grado di agevolare compiti specifici come l'interpolazione multidimensionale (SuperPolator di Erik van Blokland) o la gestione del kerning (MetricsMachine di Tal Leming). L'interscambio dei dati viene assicurato dal formato UFO e molte delle librerie su cui poggiano gli strumenti – come Vanilla o Defcon – sono distribuite con licenze *open* e mantenute dalla comunità degli sviluppatori. FontLab rimane però un importante compromesso: l'applicazione è sviluppata originariamente per Windows e non ha mai integrato l'esportazione dei suoi file in formato UFO.

Nel 2009 Frederik Berlaen, su iniziativa di David Berlow e Petr van Blokland, inizia lo sviluppo di un font editor nativo per UFO e sviluppato unicamente per OSX, presentato all'Atypi 2011 di Reykjavík. La filosofia progettuale di RoboFont è a dir poco radicale: da un lato fornisce unicamente le funzionalità base per il disegno senza affollare l'interfaccia, dall'altro permette di accedere e manipolare tutti i componenti dell'applicativo. L'utente è così incoraggiato a plasmare l'ambiente di lavoro in base alle proprie necessità. Dalla semplicità con cui è possibile integrare componenti aggiuntivi nasce nel 2013 il progetto Mechanic,² un gestore di estensioni per RoboFont ospitate su GitHub.

L'ecosistema dei RoboTools ha raggiunto la piena maturità. Il modello open source adottato ha consentito la condivisione degli oneri di sviluppo dei componenti base, permettendo ai singoli progettisti di costruire i propri strumenti su solide fondamenta. L'accesso all'ecosistema rimane più costoso delle alternative chiavi in mano sul mercato, ma molti hanno ormai colto le potenzialità di questa estrema flessibilità.

1 <http://freehandforum.org>

2 www.robofontmechanic.com



Uno screenshot della fase di disegno dell'Obsidian. L'interfaccia è realizzata mediante Vanilla e il disegno delle ombre mediante Drawbot.
— A screenshot of the Obsidian design phase. The interface is created using Vanilla and the design of the shadows with Drawbot.
© Hoefler & Co.

Il plug-in permette di illuminare con angolazioni variabili le differenti porzioni del glifo in modo da ottimizzare le ombre risultanti.
— The plugin allows you to light different portions of the glyph from varying angles in order to optimize the resulting shadows.
© Hoefler & Co.

Over the last twenty years a group of designers has systematically questioned the market's offer of *type design tools*. An invitation to regain possession of production methods combining retail licenses and open source.

In March 2015, the quiet chaos that animates the Hague Academy of Fine Arts was rippled for a couple of days. Robothon, the conference dedicated to font software and technology hosted every three years by the Academy was in full swing.

The atmosphere was that of a family reunion: just about the whole of the Netherlands and the Berlin of type design plus a more discreet delegation from the West and East coasts of the United States were there. After the usual round of pleasantries, the crowd swarmed to the auditorium, ready to nosedive into a busy programme of talks. The conference was introduced by Guido van Rossum, Python's and one of the most influential names in contemporary computing. The story began twenty years earlier, and it was made up of brothers, academy classmates and hybridization between open and retail.

By the early nineties the desktop publishing revolution had been reality for about ten years and digital fonts were still one of the founding pillars of IT democratization. There was a steady move from expensive and cumbersome digitization methods to desktop applications able to support the design of cubic Bézier curves and export to PostScript format. The most popular application among professionals was Fontographer, in production since the mid-eighties and which would lead to the famous FreeHand.¹ In an effort to combat the monopoly of Adobe's PostScript format, in 1991 Apple and Microsoft released TrueType, among whose peculiarities was the fact that it exploited quadratic Bézier curves, which cannot be converted without loss of quality or adding points from third degree curves. For companies that wanted total control over the design, it therefore became necessary to manipulate the new curves directly in the editor. At the initiative of David Berlow, founder of FontBureau, Petr van Blokland along with Letterror – his younger brother Erik and Just van Rossum – distributed RoboFog, a version derived from Fontographer, in 1996.

As a font design proceeds, the more repetitive tasks exponentially become: whether it be the positioning of diacritics or interpolation of multiple masters, developing an interface to perform such different tasks soon

turns out to be impracticable. Hence the idea of creating a gateway to the core functionality of the application by using a scripting language. After a first attempt with AWK, Just suggested giving a chance to the open source language developed by his elder brother Guido, the first version of which was dated January 1994. It was a happy union, destined to radically influence font editing. Indeed, designed for a public of non-programmers, Python had a very neat and accessible syntax at the expense of calculation speed. RoboFog was developed up until 2001, when the old Fontographer source code proved unable to survive the transition to OSX.

As early as 1998 FontLab developer Yuri Yarmola had integrated Python into the 2.0 version of the application by releasing a turnkey Python API (Application Program Interface). When RoboFog stopped being developed and its users migrated to FontLab, with the help of Tal Leming Letterror used this API to transfer the RoboFog object model to the new application. The development of this module started in 2003 and the result was presented a year later under the name RoboFab. The module introduced a new font format called UFO (Unified Font Object), now standard format for exchanging font data. The design philosophy behind it is summed up in the form of a manifesto:

1. The data must be human readable and human editable.
2. The data should be application independent.
3. Data duplication should be avoided unless absolutely necessary.

At the beginning of the twenty-first century, the market was dominated by FontLab which, however, could not keep up with the new design needs. FontLab was designed as a generalized application rather than for a specialized audience. Meanwhile, some of the designers mentioned earlier continued in the development of applications that facilitated specific tasks, such as multi-dimensional interpolation (SuperPolator by Erik van Blokland) or kerning (MetricsMachine by Tal Leming). Data exchange is ensured by the UFO format and many of the libraries the tools rely on – such as Vanilla or



Defcon – are distributed with open licenses and maintained by the developer community. FontLab remains an important compromise though: the application was originally developed for Windows and it has never integrated the export of its files in UFO format.

Prompted by David Berlow and Petr van Blokland, in 2009 Frederik Berlaen began the development of a font editor native for UFO and developed solely for OSX. It was presented at Atypi 2011 in Reykjavik. The design philosophy of RoboFont is nothing short of radical: on the one hand it provides only the basic functions for the design without cluttering the interface, while on the other it allows you to access and manipulate all the components of the application. In this way users are encouraged to shape the work environment to suit their needs. The ease with which you could integrate additional components led in 2013 to the Mechanic² project, a manager of extensions for RoboFont, available on GitHub.

The ecosystem of RoboTools has reached full maturity. The open source model adopted has enabled development costs of the basic components to be shared, allowing individual designers to build their own instruments on firm foundations. Access to the ecosystem remains more expensive than other turnkey alternatives on the market, but many have now grasped the potential of this extreme flexibility.



Workshop di Python tenuto da Just van Rossum e Frederik Berlaen durante Robothon15.
— Python workshop held by Just van Rossum and Frederik Berlaen at Robothon15.
Photo © Erik van Blokland

Da sinistra verso destra — Left to right:
Frederik Berlaen, Ben Kiel,
Just van Rossum, Tal Leming.
Photo © Erik van Blokland

¹ <http://freehandforum.org>

² www.robofontmechanic.com

Description / Descrizione	License / Licenza	Developers / Sviluppatori
Font Editor / Font Editor	Retail	Frederik Berlaen
Linear interpolation app / Applicazione per l'interpolazione lineare	Retail	Frederik Berlaen
OpenType features previewer and debugger / Gestore di features OpenType	MIT	Tal Leming
UFO based objects application oriented / Modello a oggetti UFO orientato allo scripting	MIT	Tal Leming
Pythonic wrapper for Cocoa components / Involucro Python per componenti Cocoa	MIT	Tal Leming, Just van Rossum
Library for calculations on font data / Libreria per calcolo su dati font	MIT	
Interpolation compatibility manager / Gestore della compatibilità dei tracciati ai fini dell'interpolazione	Retail	Tal Leming
Multidimensional interpolation manager / Applicazione per l'interpolazione multidimensionale	Retail	Erik van Blokland
Library for linear multidimensional interpolation / Libreria per la gestione dell'interpolazione multidimensionale	BSD	Erik van Blokland
2D graphics generator / Generatore di grafica 2D	BSD	Just van Rossum, Erik van Blokland, Frederik Berlaen
Library for manipulating data associated with fonts / Libreria per la manipolazione di dati font	BSD	Tal leming, Erik van Blokland, Just van Rossum
Font kerning editor / Gestore dei dati di crenatura	Retail	Tal leming
Library for manipulating SFNT based fonts / Libreria per la gestione di font SFNT	BSD	Just van Rossum (1999-2013) Behdad Esfahbod (2013-)
Font editor / Font editor	Retail	Petr van Blokland, Just van Rossum, Erik van Blokland

Timeline delle librerie e degli applicativi
dell'ecosistema RoboTools.
— Timeline of libraries and applications
of the RoboTools ecosystem.
© Roberto Arista

